CpE 645 Image Processing and Computer Vision

Prof. Hong Man

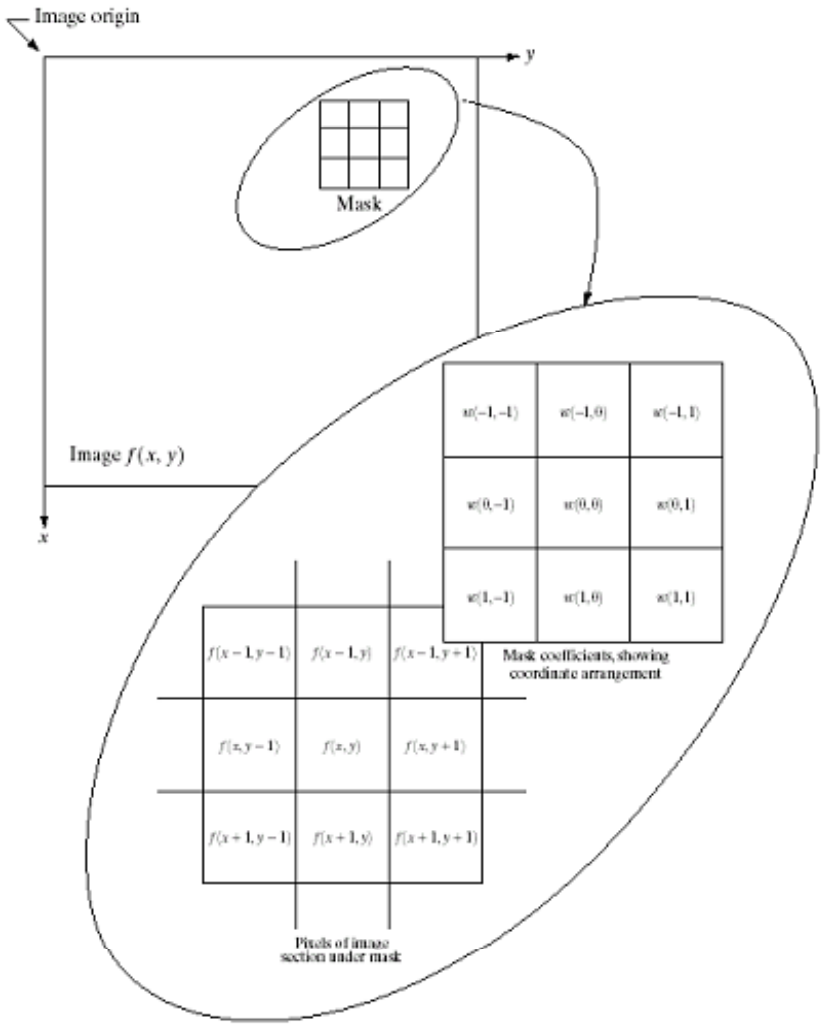
Department of Electrical and Computer Engineering
Stevens Institute of Technology

STEVENS Institute of Technology

Visual Information Environment Laboratory

Area Processing

- Area processing usually are 2-D filtering operation.
- Two major tasks:
  - Smoothing: removing noise
  - Sharpening: enhance edges
- These two operations have opposite effects.



Smoothing and Sharpening Examples

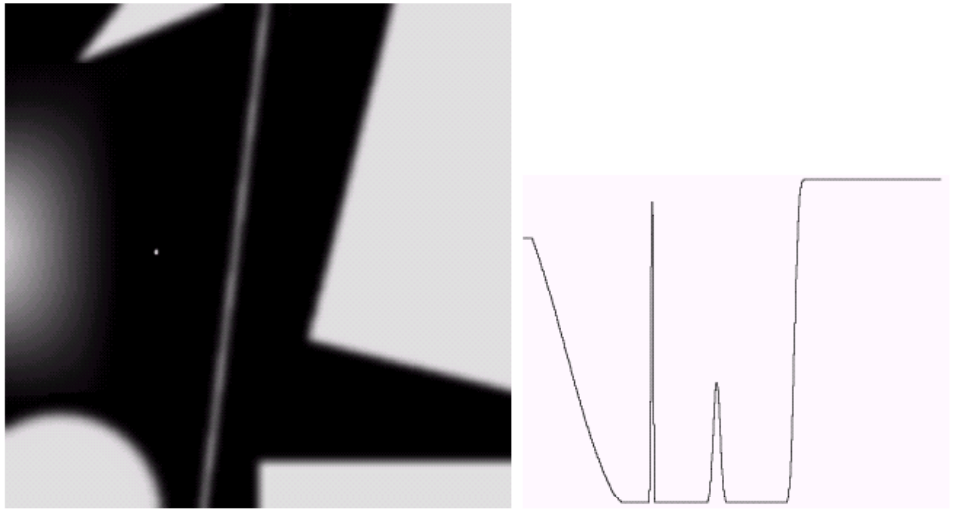
Smoothing

Sharpening

Image Study

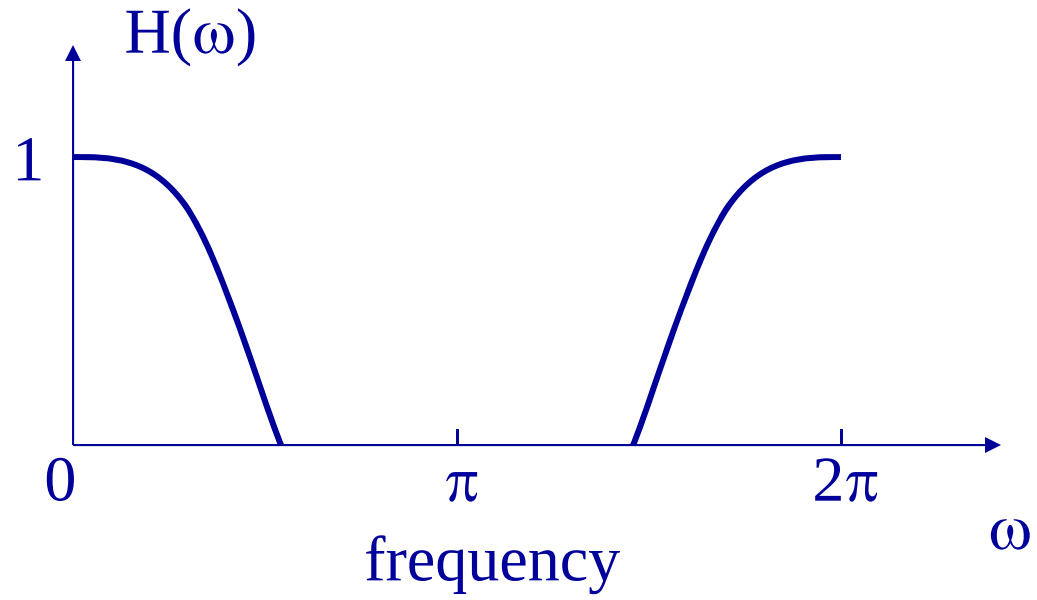
a b
c

FIGURE 3.38
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point.
(c) Simplified profile (the points are joined by dashed lines to simplify interpretation).



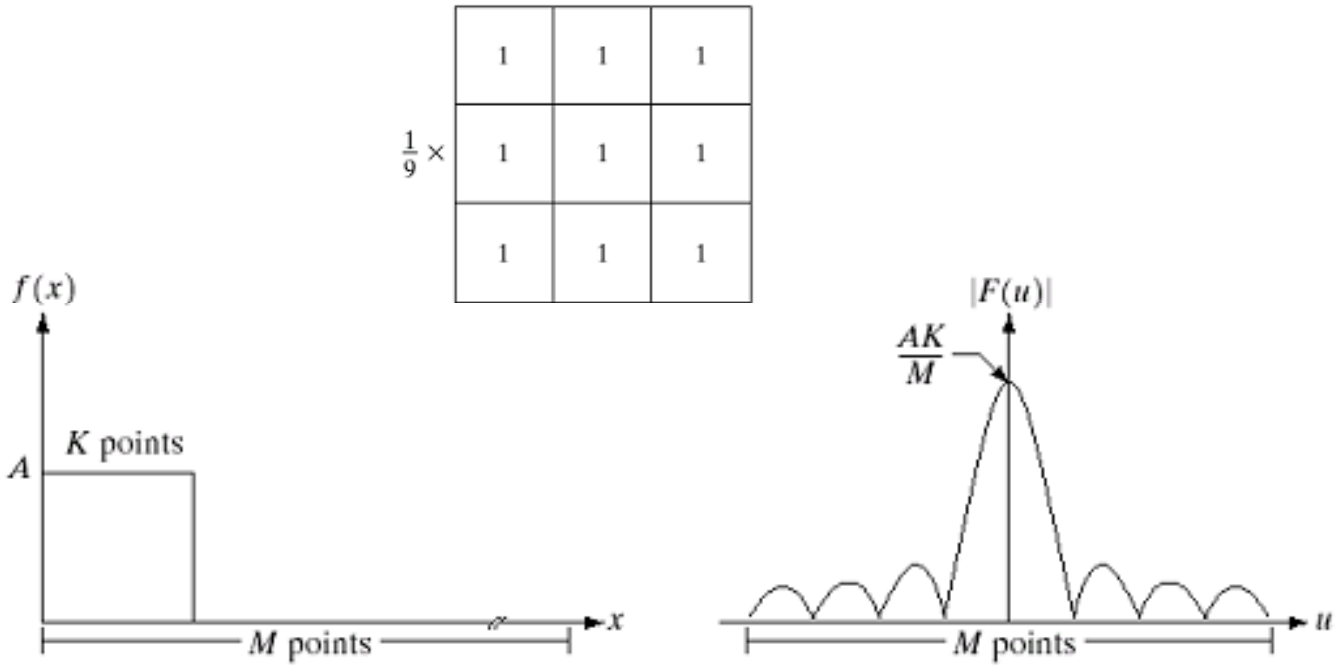
Smoothing

- Smoothing is to remove small isolated noise and unwanted details.
- Smoothing is generally achieved through low-pass filtering.



Smoothing

- A simple low-pass filter is the local averaging operator.



Smoothing

original

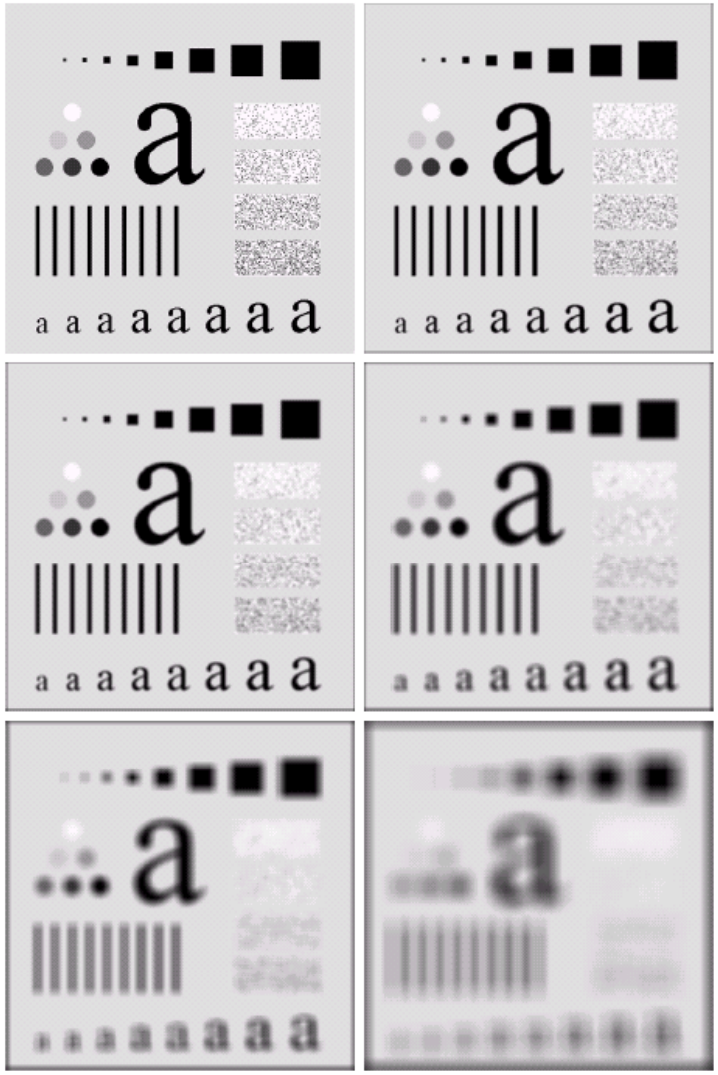
averaging mask = 3x3

averaging mask = 5x5

averaging mask = 9x9

averaging mask = 15x15

averaging mask = 35x35



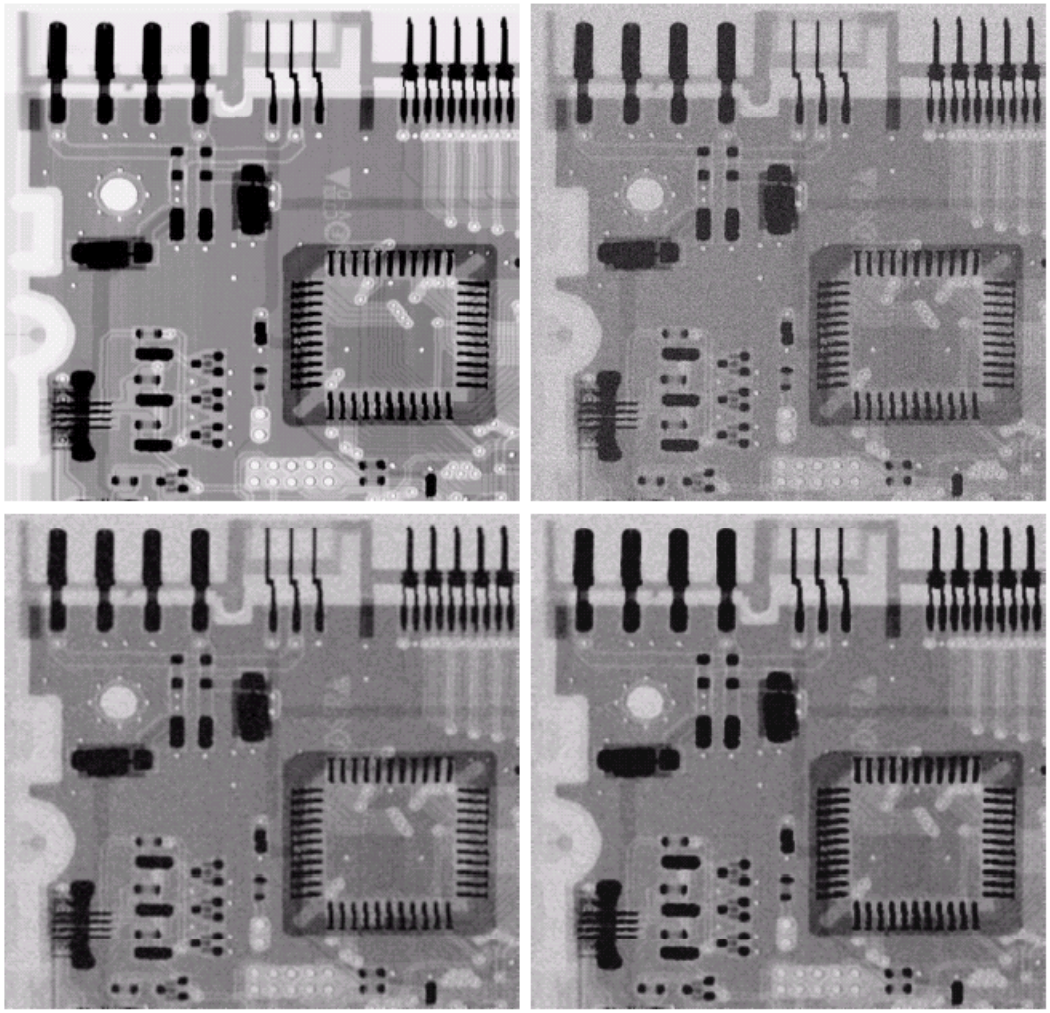
General Mean Filters

- Assume $S[n_1, n_2]$ represent a set of image coordinates within a given rectangular subimage window with size of $M \times N$ centered at $[n_1, n_2]$.
- *Arithmetic mean filter* reduces noise as a result of blurring.
- *Geometric mean filter* smoothes better than arithmetic mean filter, yet lose less image detail in the process.

$$\text{Arithmetic mean filter} \quad \hat{f}[n_1, n_2] = \frac{1}{MN} \sum_{[m_1, m_2] \in S[n_1, n_2]} g[m_1, m_2].$$

$$\text{Geometric mean filter} \quad \hat{f}[n_1, n_2] = \left[\prod_{[m_1, m_2] \in S[n_1, n_2]} g[m_1, m_2] \right]^{\frac{1}{MN}}.$$

General Mean Filters



a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3×3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

General Mean Filters

- *Harmonic mean filter* works well with salt noise, but fails with pepper noise. It does well on Gaussian noise.

Harmonic mean filter

$$\hat{f}[n_1, n_2] = \frac{MN}{\sum_{[m_1,m_2]\in S[n_1,n_2]} \frac{1}{g[m_1, m_2]}}.$$

General Mean Filters

- *Contraharmonic mean filter* can virtually eliminate salt and pepper noise. Positive Q is used for eliminating pepper noise, and negative Q is used for salt noise. When $Q = 0$, it becomes arithmetic mean filter, and when $Q = -1$, it is harmonic mean filter.

Contraharmonic mean filter

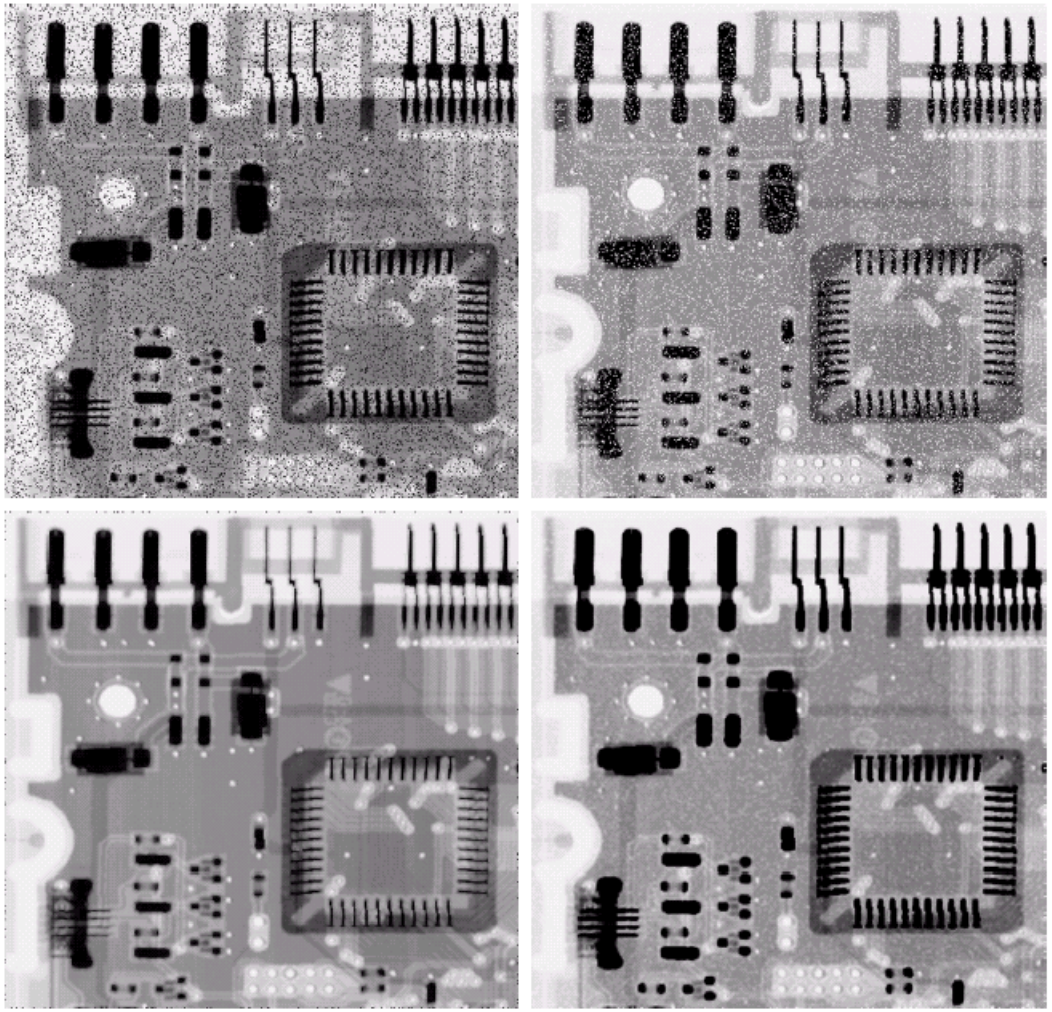
$$\hat{f}[n_1, n_2] = \frac{\sum_{[m_1,m_2] \in S[n_1,n_2]} g[m_1, m_2]^{Q+1}}{\sum_{[m_1,m_2] \in S[n_1,n_2]} g[m_1, m_2]^{Q}}$$

where Q is the order of the filter .

General Mean Filter

a b
c d

FIGURE 5.8
(a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.



General Mean Filter

- If we delete the $d/2$ lowest and $d/2$ highest gray values of $g[n_1, n_2]$ in the $S[n_1, n_2]$ window, and let $g_r[n_1, n_2]$ represent the remaining $M \times N - d$ pixels, the resulting arithmetic mean filter is *Alpha-trimmed mean filter*. It works well with multiple types of noise, such as combination of salt and paper and Gaussian noise.

Alpha - trimmed mean filter

$$\hat{f}[n_1, n_2] = \frac{1}{MN - d} \sum_{[m_1, m_2] \in S[n_1, n_2]} g_r[m_1, m_2].$$

General Mean Filter

(a) Image corrupted by additive uniform noise,

(b) Image (a) corrupted by additional salt&pepper noise.

(c) Arithmetic mean filter

(d) Geometric mean filter

(e) Median filter

(f) Alpha-trimmed mean filter.

All filters are with size of 5×5.

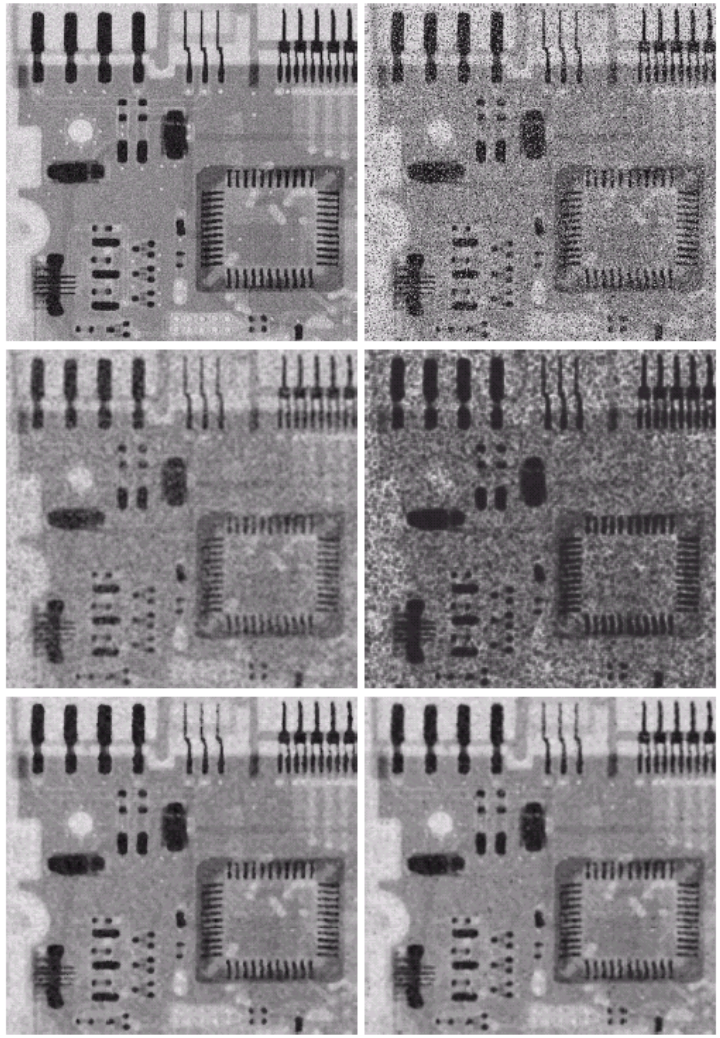


Image Averaging

- There will always be noise introduced in the imaging process, e.g. sensor (CCD) noise.
- If we can obtain multiple images of the same scene, we can average all these images to reduce noise.
- Consider a noisy image $g[n_1, n_2]$ formed by the original scene $f[n_1, n_2]$ with some additive noise $\eta[n_1, n_2]$

$$g[n_1, n_2] = f[n_1, n_2] + \eta[n_1, n_2]$$

assuming noise term $\eta[n_1, n_2]$ is zero mean and uncorrelated.

- If multiple noisy images are obtained

$$g_i[n_1, n_2] = f[n_1, n_2] + \eta_i[n_1, n_2]$$

for $1 < i \leq K$

Image Averaging

- The averaged image is calculated as

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^{K} g_i(x, y)$$

it follows that

$$E\{\bar{g}[n_1, n_2]\} = f[n_1, n_2], \text{ and } \sigma^2_{\bar{g}[n_1,n_2]} = \frac{1}{K} \sigma^2_{\eta[n_1,n_2]}$$

where $E\{\bar{g}[n_1, n_2]\}$ is the expected value of $\bar{g}[n_1, n_2]$ and $\sigma^2_{\bar{g}[n_1,n_2]}$, $\sigma^2_{\eta[n_1,n_2]}$ are the variances of $\bar{g}$ and η at position $[n_1, n_2]$.

The standard deveration at any point in the averaged image will be $\sigma_{\bar{g}[n_1,n_2]} = \frac{1}{\sqrt{K}} \sigma_{\eta[n_1,n_2]}$

Image Averaging

- We can see that by repeatedly taking the images of the same scene, and averaging them, we can reduce the noise variation at a factor of $\frac{1}{\sqrt{K}}$.

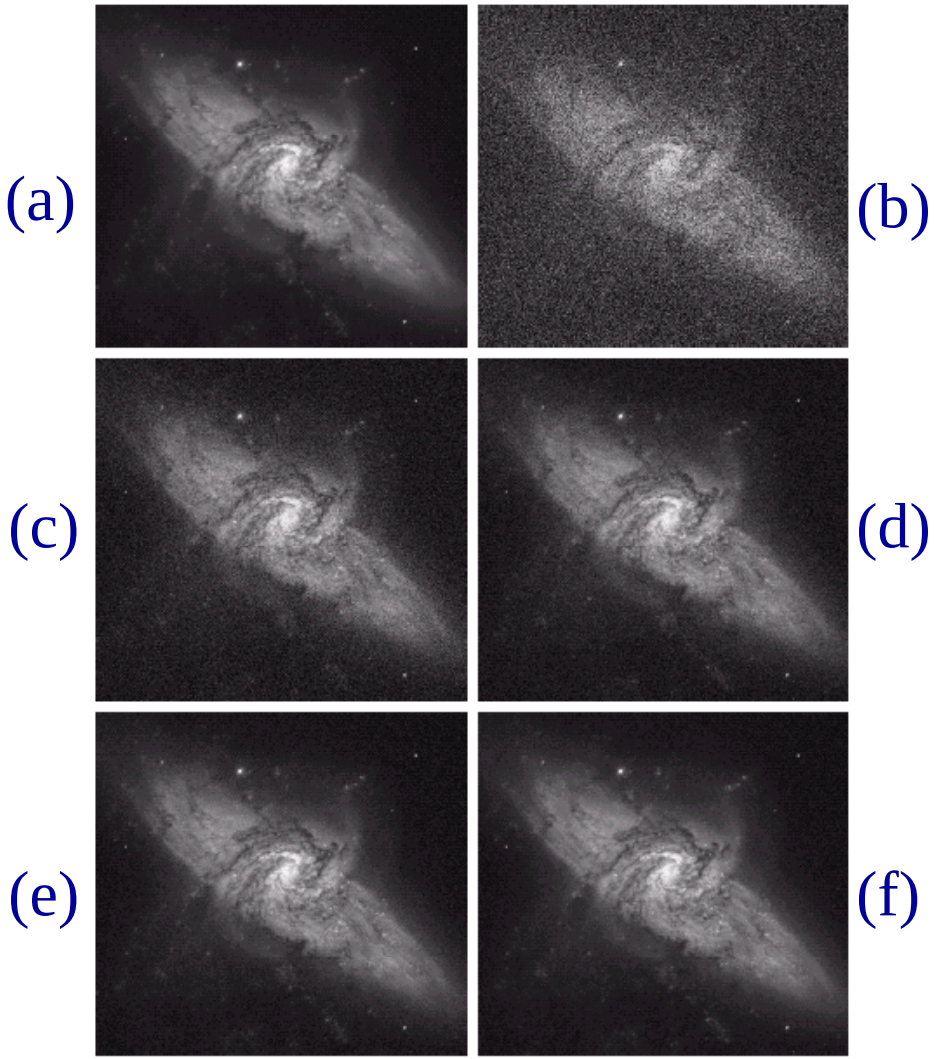
- Image averaging has to be carefully conducted with proper image registration (alignment of objects).
- Example of image averaging can be found in the field of astronomy, where scenes are relatively invariant, and images are taken over long period of time.

Image Averaging

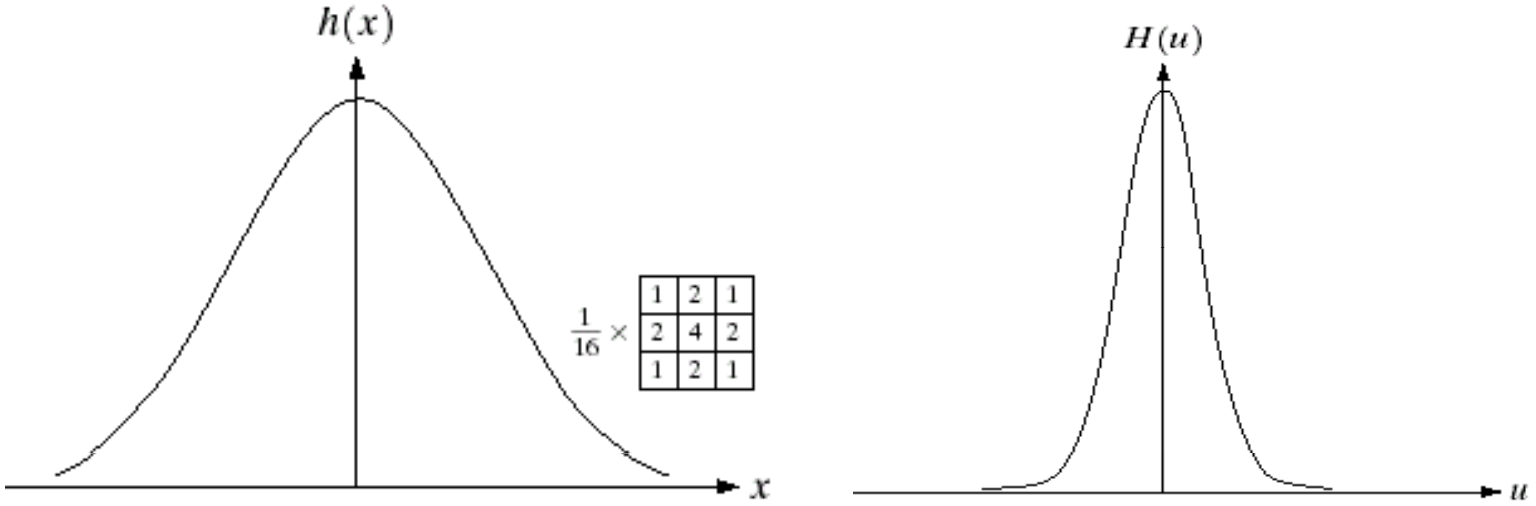
(a) Image of Galaxy Pair NGC 3314,

(b) Image corrupted by additive Gaussian noise with zero mean and a standard variation of 64 gray-levels.

(c) - (f) results of averaging K = 8, 16, 64 and 128 noisy images.



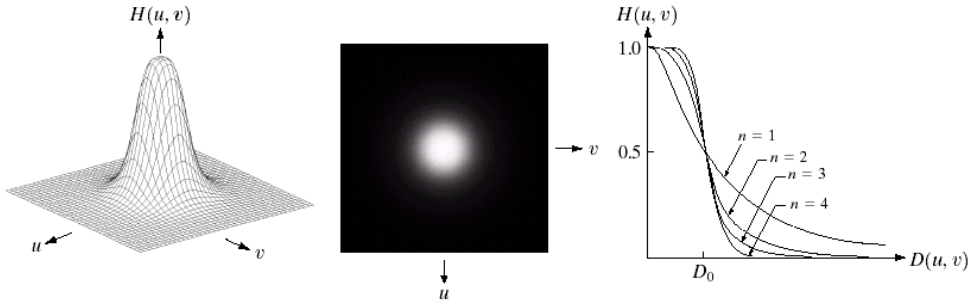
Low Pass Filtering



- There are many other low-pass filters (LPFs) that can be used for smoothing.
- Low-pass filters generally have smooth variation in spatial domain.

Low Pass Filtering

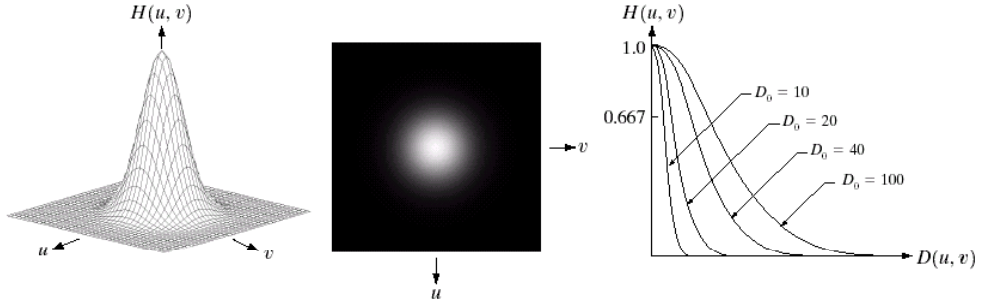
Butterworth low pass filters in frequency domain



a b c

FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Gaussian low pass filters in frequency domain



a b c

FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

Low Pass Filtering

- Usually we use symmetric, finite impulse response (FIR) filters with size of 3×3 and 5×5.
  - Small size filters produce less artifacts.
  - Odd number sizes introduces integer shift.
  - Symmetric filters are generic, convenient.
- To avoid signal attenuation or amplification after filtering, the summation of all filter coefficients is usually chosen to be 1.
- In general, don't use filter size large than 5×5 to avoid over smoothing.

Directional Low Pass Filtering

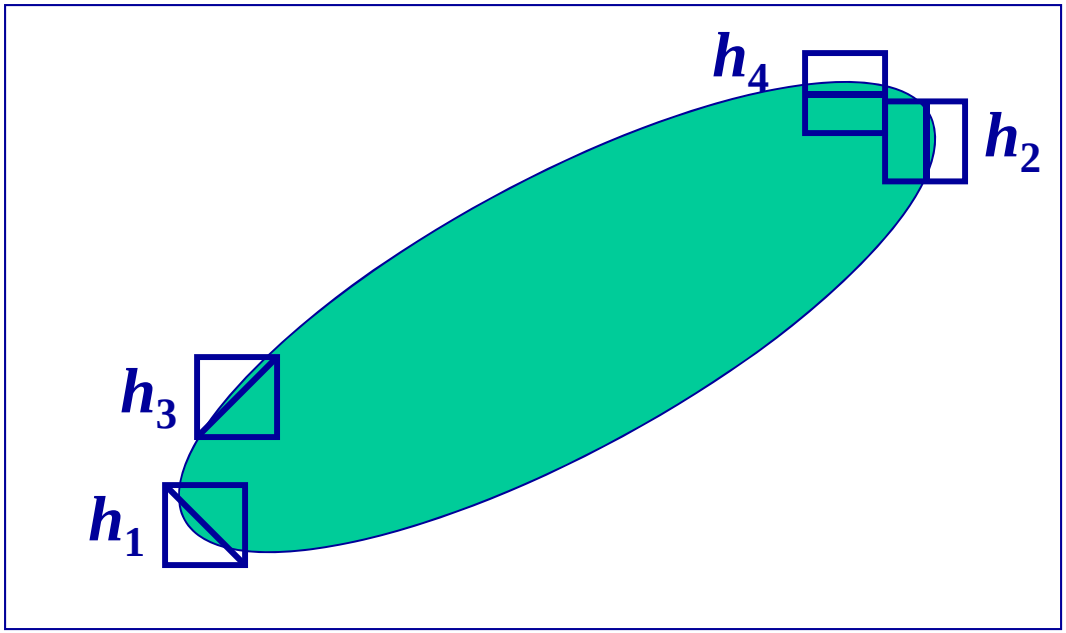
- Directional filters are low-pass filters for noise reduction.
- In order not to produce blurred edges, these filters try to avoid filtering across object boundaries. This can be achieved by filtering only along the direction with minimum amplitude variation.
- Example:

$$h_1 : \frac{1}{3}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, h_2 : \frac{1}{3}\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, h_3 : \frac{1}{3}\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, h_4 : \frac{1}{3}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

These filters all take the average of three samples.

Directional Low Pass Filtering

- One can adaptively select one of these filters according to local direction of minimum amplitude variance.
- This approach is effective for removing white noise.



Adaptive Filtering

- In general adaptive filtering attempts to adjust filter parameters according to local input data (image) behavior.
- A simple adaptive local noise reduction filter is defined as
  - The variance of overall noise, σ_η^2, is first estimated,
  - Filter operates on sliding window $S[n_1, n_2]$,
  - At each $[n_1, n_2]$, local subimage variance σ_S^2 and local mean M_s is calculated.
  - The filter output will be

$$\hat{f}[n_1, n_2] = g[n_1, n_2] - \frac{\sigma_\eta^2}{\sigma_S^2}\left(g[n_1, n_2] - M_S\right).$$

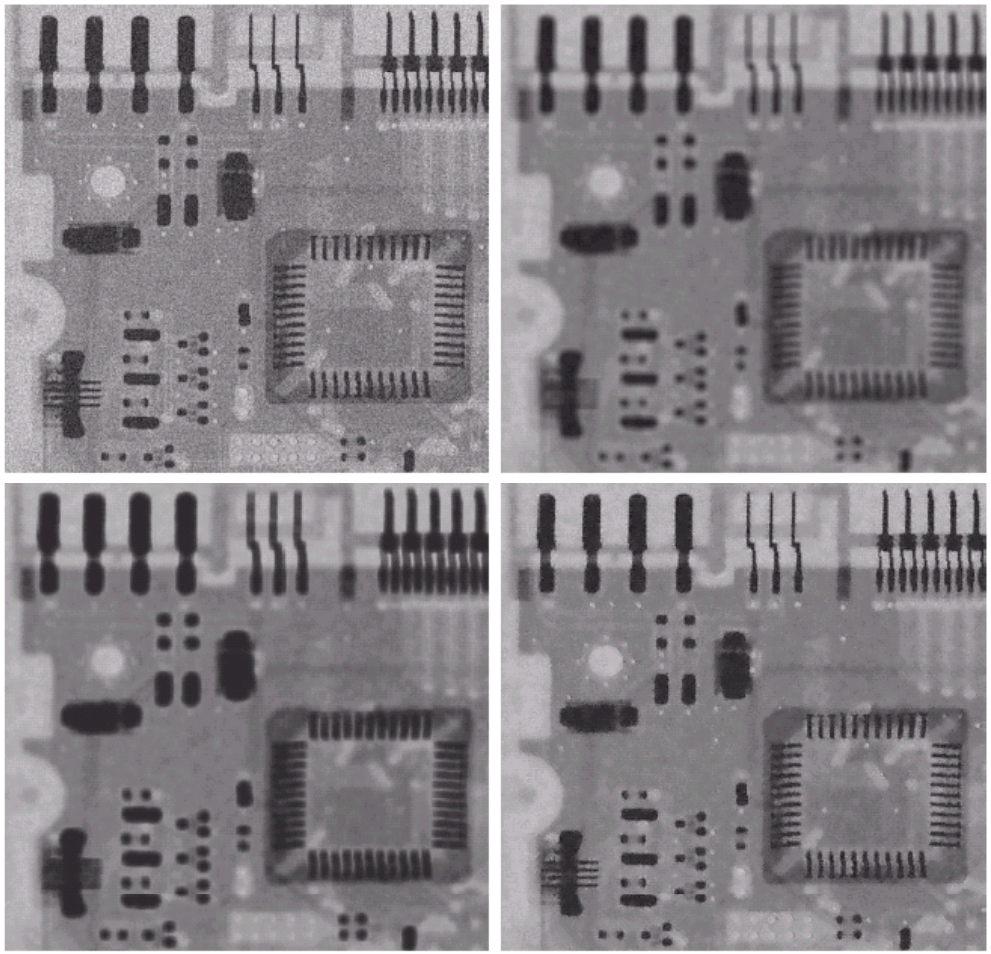
Adaptive Filtering

- The effects of this adaptive filtering are:
  - If σ_η^2 is very low, the output pixel value will be the same as the input pixel value.
  - If σ_S^2 is very high, which indicates local activity can be caused by edges and should be preserved, the output pixel value will again be the same as the input pixel value.
  - If $\sigma_\eta^2 = \sigma_S^2$, the output pixel value will be the arithmetic mean M_S.
  - In all other situations, the output value will be a weighted sum of the input value and the arithmetic mean.

Adaptive Filtering

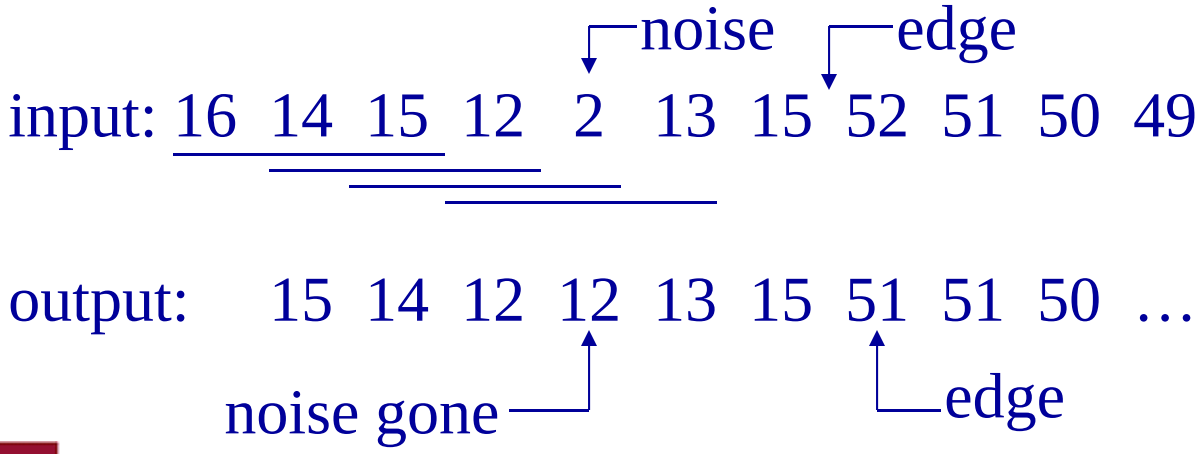
a b
c d

FIGURE 5.13
(a) Image corrupted by additive Gaussian noise of zero mean and variance 1000.
(b) Result of arithmetic mean filtering.
(c) Result of geometric mean filtering.
(d) Result of adaptive noise reduction filtering. All filters were of size 7×7.

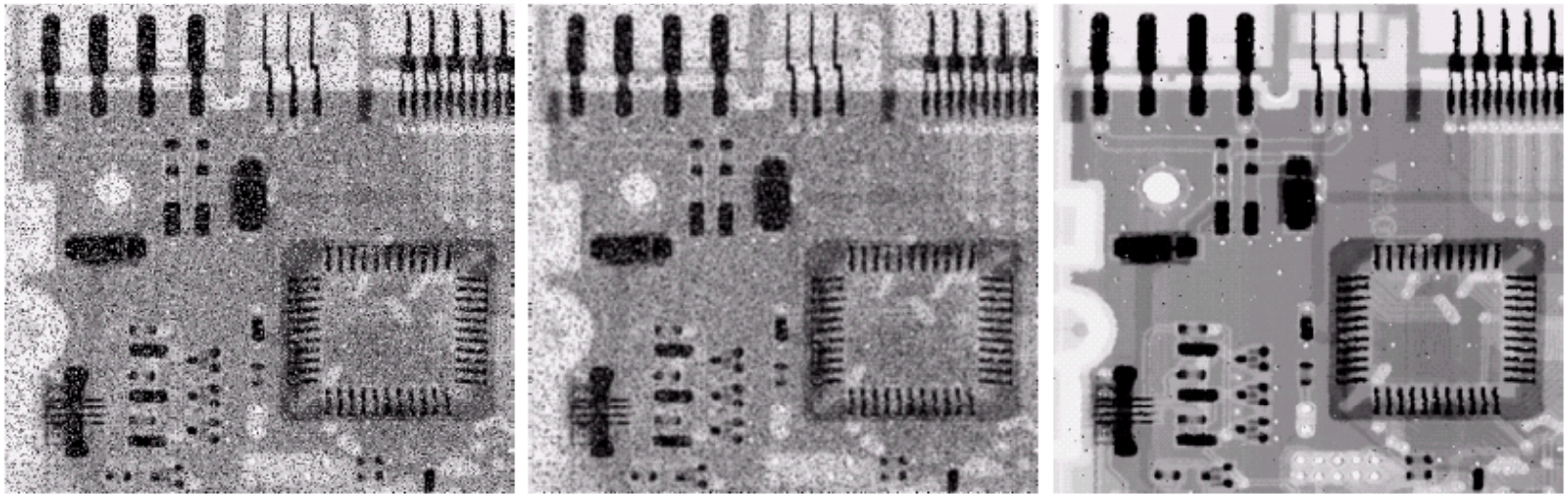


Median Filtering

- Problem with low-pass filtering: blurring edges and other sharp details.
- Median filtering: amplitude value of each pixel is replaced by the **median** of the amplitude values within the ***M*×*M*** mask. Median filtering is a **non-linear** operation.
- Example: a 1-D median filter with length of 3



Median Filtering

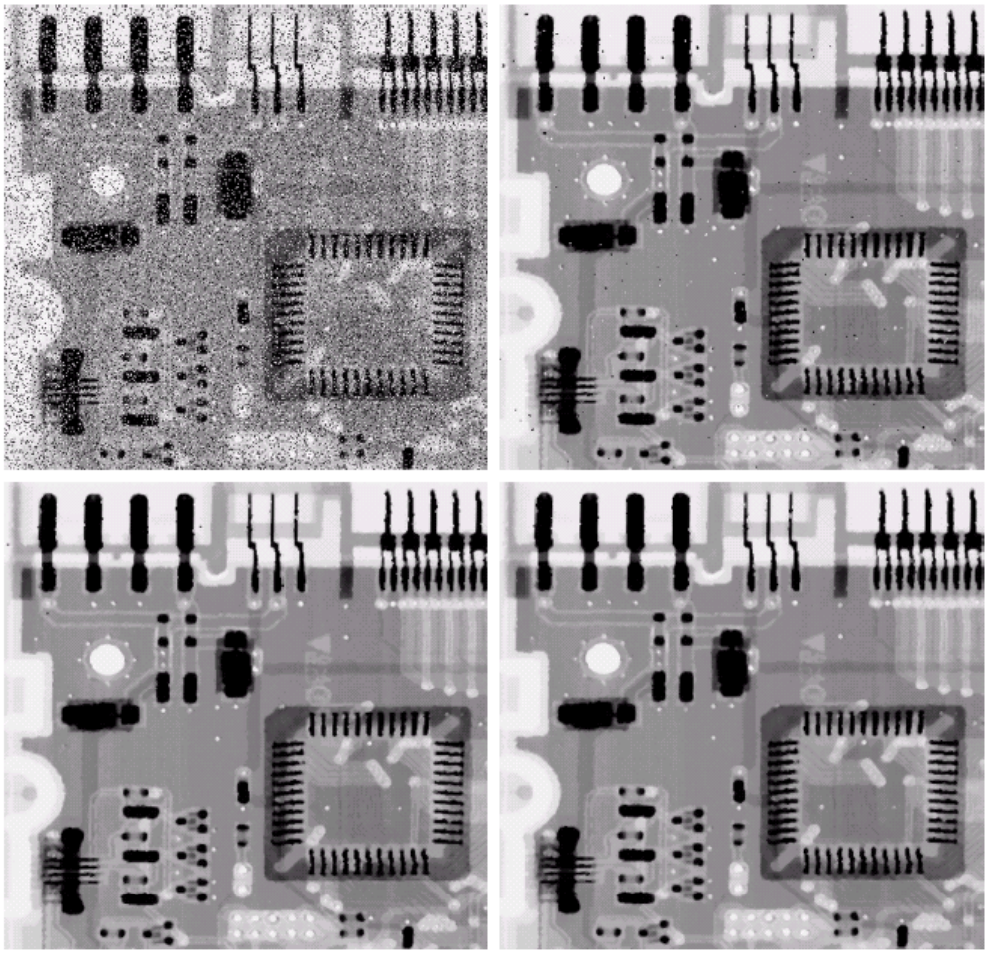


original 3x3 averaging 3x3 median

Median Filtering

a b
c d

FIGURE 5.10
(a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$.
(b) Result of one pass with a median filter of size 3×3.
(c) Result of processing (b) with this filter.
(d) Result of processing (c) with the same filter.



Median Filtering

Salt pepper noise

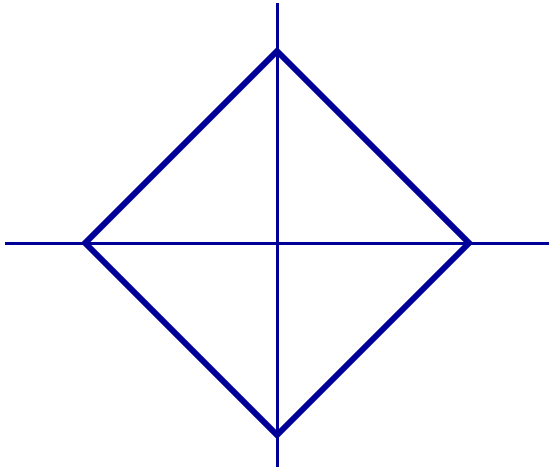
De-noise

STEVENS Institute of Technology

Visual Information Environment Laboratory

Median Filtering

- Median Filter is fully defined by its window.
- A 2-D separable median filter can be formed by applying 1-D median filters to the rows and then to the resulting columns.
- A 2-D non-separable median filter can also be formed by defining a 2-D non-separable window, e.g.



Median Filtering

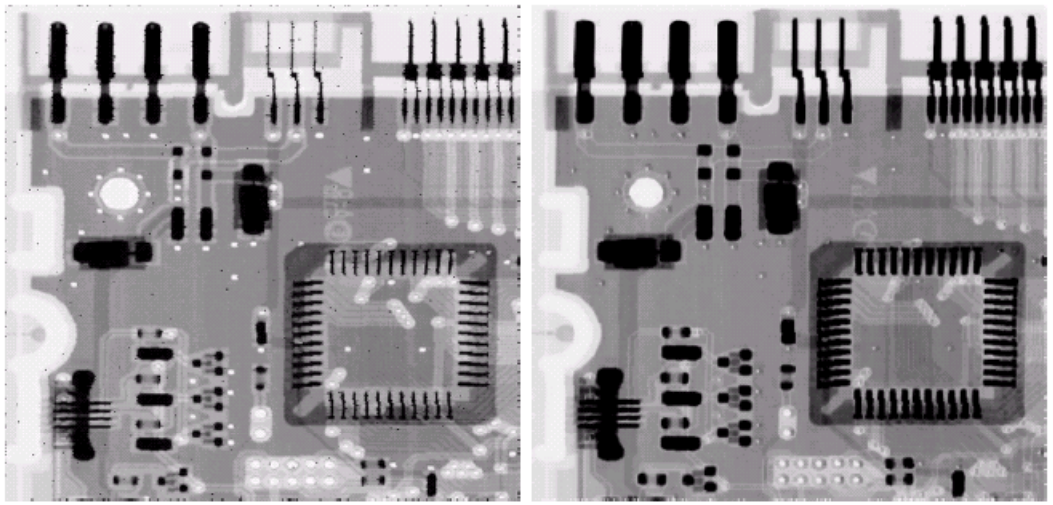
- The typical window sizes are **3×3**, **5×5** etc. The larger the window is, the more sever the smoothing effect will be.
- Median Filters are suitable for "salt and pepper" noise in gray images and speckle noise in binary images.
- Median filters are a subset of the general rank order filters
- A rank order filter is defined by its window and the rank of the output. Assume a rank order filter has a window with size of N. The M^{th}-rank filter ($M \leq N$) associated with this window selects the M^{th} smallest value as the output.

  e.g. first-rank filter – minimum value output – min filter,

  N^{th}-rank filter – maximum value output – max filter.

Rank Order Filtering

- Min filter generally captures local darkest points, and max filter can captures local brightest points.

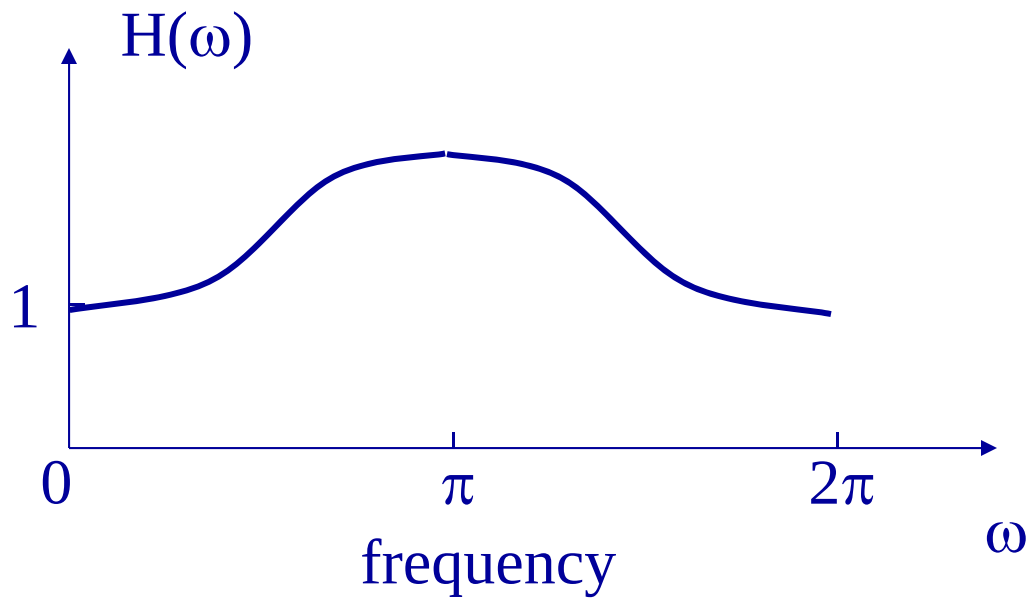


a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

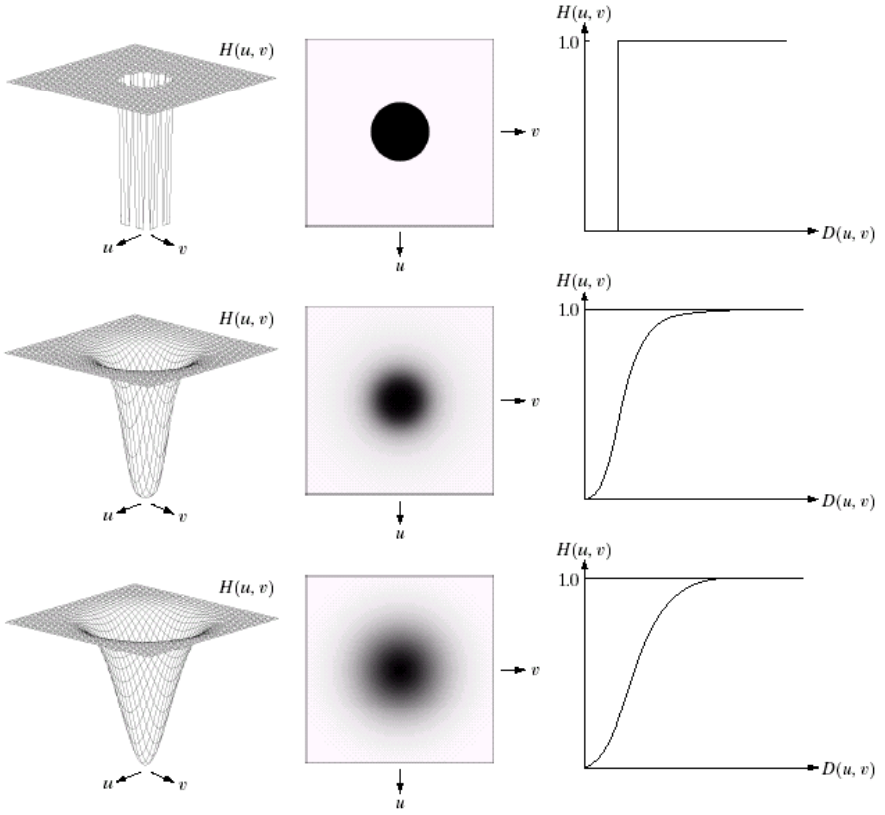
Sharpening

- Sharpening is to enhance the details and edges of an image.
- Sharpening usually is achieved through high frequency emphasis (or high-boost) filtering.



Sharpening

Some high pass filters in frequency domain.



a b c
d e f
g h i

FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Sharpening

- A simple high-boost filter can be obtained through a low pass filtering, subtraction, and summation, this process is called *unsharp masking*.

$$\mathbf{Y = \lambda \{X - LPF(X)\} + X}$$

where **X** is the original image,

LPF(·) is a low-pass filtering,

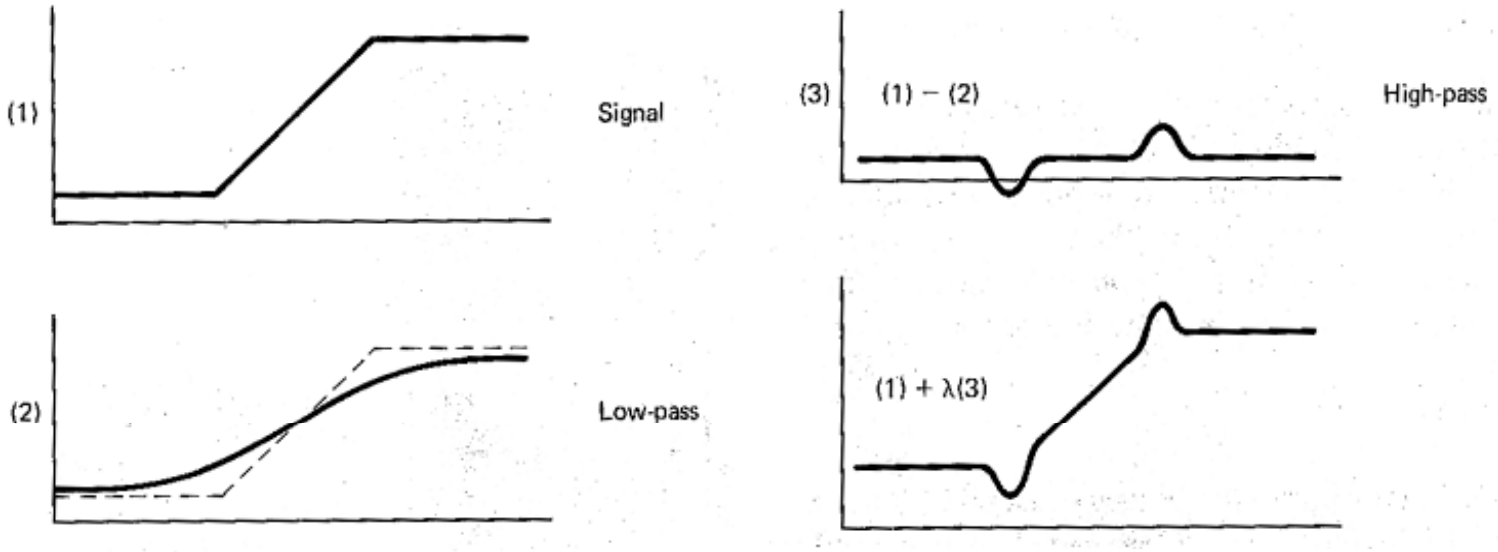
X – LPF(X) effectively forms a high-passed version of the original **X**,

λ is a constant,

Y is the high frequency emphasized image.

Sharpening

- A high-boost filtering example



Derivates

- First and second order derivates can be used as sharpening operators.
- 1-D derivates

  first order derivate $\frac{\partial f}{\partial x} \approx f(x+1) - f(x)$

  second order derivate $\frac{\partial^2 f}{\partial x^2} \approx f(x+1) + f(x-1) - 2f(x)$

- 2-D second order derivate

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\approx f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x)$$

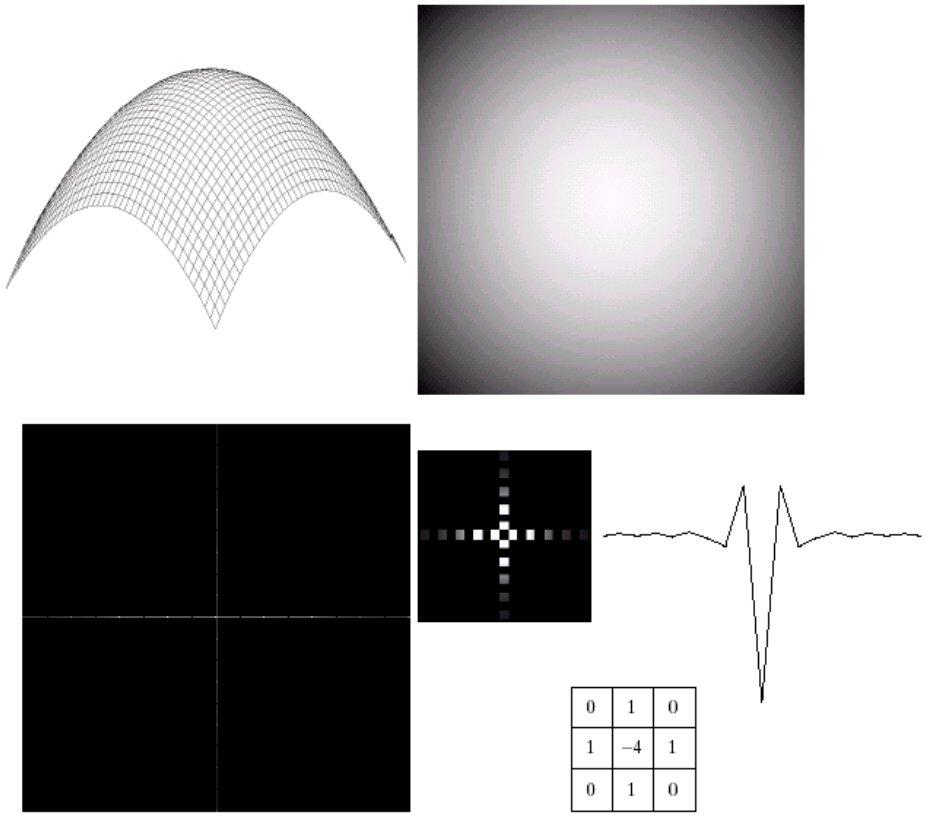
Laplacian Operators

- *Laplacian operator* is a simple 2nd order derivate operator.
- There are several discrete forms of the Laplacian operator.
- Laplacian operators are commonly used as high-pass filters.
- These operators can be used to form the high-boost filters for image sharpening.

0	1	0	1	1	1
1	−4	1	1	−8	1
0	1	0	1	1	1
0	−1	0	−1	−1	−1
−1	4	−1	−1	8	−1
0	−1	0	−1	−1	−1

Laplacian Operators

Laplician operator in frequency domain.



a b
c d e
f

FIGURE 4.27 (a) 3-D plot of Laplacian in the frequency domain. (b) Image representation of (a). (c) Laplacian in the spatial domain obtained from the inverse DFT of (b). (d) Zoomed section of the origin of (c). (e) Gray-level profile through the center of (d). (f) Laplacian mask used in Section 3.7.

Laplacian Operators

- Procedure to use Laplacian operators for sharpening:

$$\mathbf{Y = Laplacian(X) + X}$$

where **X** is the original image,

Laplacian(·) is a Laplacian operator – High-pass filter

Y is the sharpened image.

- This expression can be simplified as

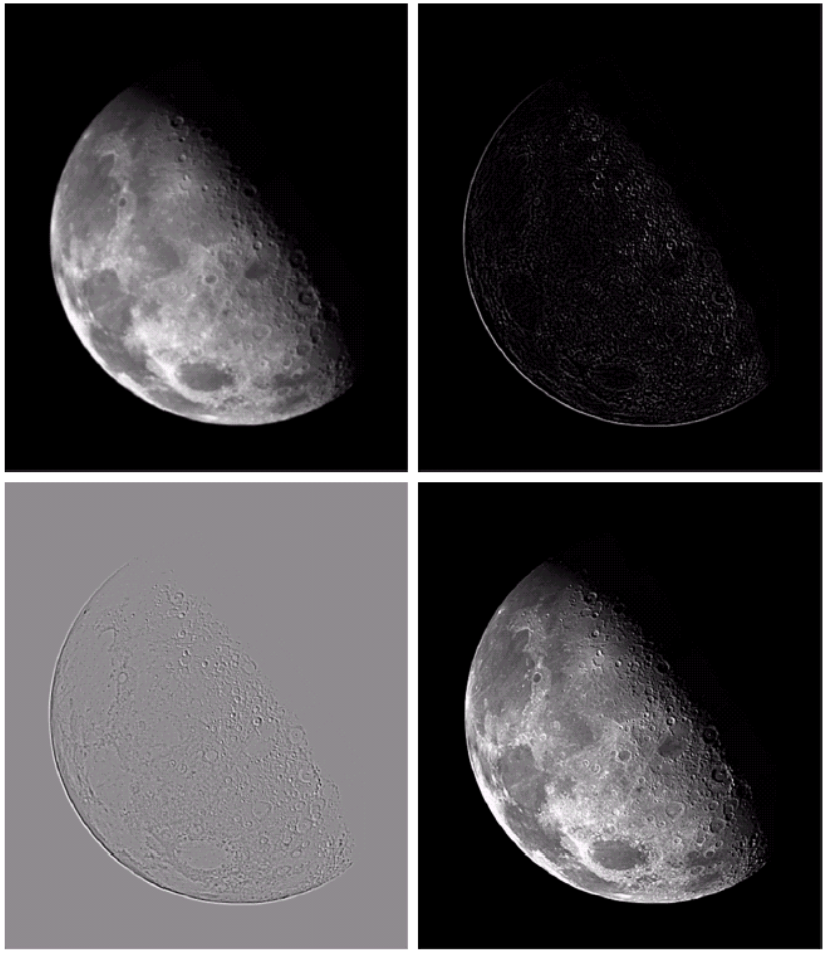
$$\mathbf{Y = Composite\_Laplacian(X)}$$

where the Composite Laplacian integrates the Laplacian operator with the simple summation operator.

Laplacian Operators

a b
c d

FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)



Composite Laplacian Operators

- For Laplacian operators

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

their composite Laplacian operators are

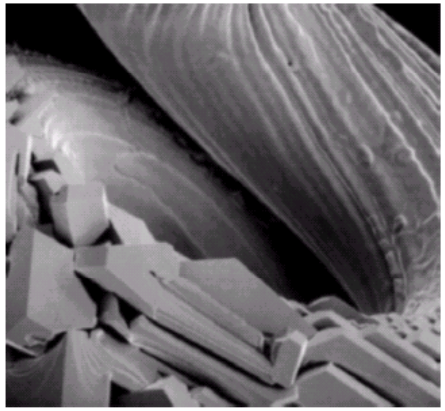
0	−1	0
−1	5	−1
0	−1	0

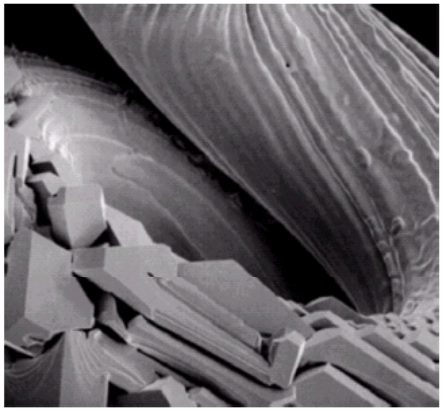
−1	−1	−1
−1	9	−1
−1	−1	−1

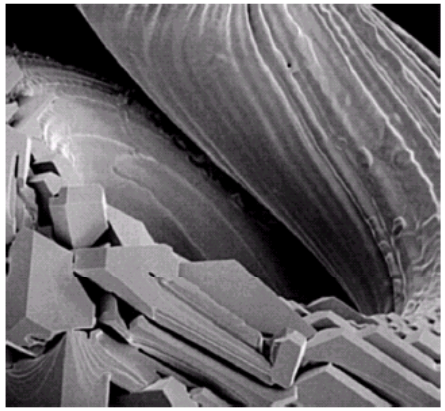
Composite Laplacian Operators

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1







a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)